

Solving chemical master equations by adaptive wavelet compression[☆]

Tobias Jahnke^{*,a}, Tudor Udrescu^a

^a*Karlsruhe Institute of Technology (KIT), Fakultät für Mathematik, Institut für Angewandte und Numerische Mathematik, Kaiserstr. 93, 76133 Karlsruhe, Germany*

Abstract

Solving chemical master equations numerically on a large state space is known to be a difficult problem because the huge number of unknowns is far beyond the capacity of traditional methods. We present an adaptive method which compresses the problem very efficiently by representing the solution in a sparse wavelet basis that is updated in each step. The step-size is chosen adaptively according to estimates of the temporal and spatial approximation errors. Numerical examples demonstrate the reliability of the error estimation and show that the method can solve large problems with bimodal solution profiles.

Key words: chemical master equation, wavelet compression, adaptive step-size selection, adaptive Galerkin approximation, Rothe's method, stochastic reaction kinetics, gene regulatory networks

PACS: 02.60.Lj, 02.70.Dh, 05.10.Gg, 82.39.Rt, 87.10.Ed, 87.18.Vf

1. Introduction

Stochastic models provide a better understanding of many complex systems in physics, chemistry, biology, ecology and other sciences. The evolution of such systems is often driven by the random interaction of d different types of particles which, depending on the applications, can represent molecules, humans, animals, or other discrete units. In nearly all processes in nature, the particle numbers are subject to random fluctuations caused by inherent stochastic noise. If all species are present in abundance, the effects of fluctuations and the discreteness of individual particles can be neglected. In this case the dynamics of the system can be reasonably modelled with the traditional reaction-rate approach, i.e. by solving a set of d ordinary differential equations for the concentrations of the species. This simplification, however, is inappropriate if applications such as, e.g., the transcription of genetic information in a gene regulatory network are investigated. Here, the evolution of the system must be regarded as a Markov jump process $\{X(t), t \geq 0\}$

[☆]Supported by the "Concept for the Future" of Karlsruhe Institute of Technology (KIT) within the framework of the German Excellence Initiative, and the DFG priority programme SPP 1324 "Mathematische Methoden zur Extraktion quantifizierbarer Information aus komplexen Systemen".

*Corresponding author

Email addresses: jahnke@kit.edu (Tobias Jahnke), udrescu@kit.edu (Tudor Udrescu)

on the d -dimensional discrete state space \mathbb{N}^d . Each state $x \in \mathbb{N}^d$ is a vector of particle numbers, and every reaction event induces a jump of the random variable $X(t)$ to a new state.

The stochastic behavior allows the reproduction of important effects of real-life systems, but often causes severe computational problems. Realizations of the Markov jump process can be generated by stochastic simulation (cf. [13]), but the main object of interest is usually the probability distribution $p(t, x) = \mathbb{P}(X(t) = x)$, and approximating this distribution up to the desired accuracy by generating a huge number of realizations can be computationally inefficient. An alternative approach is to determine $p(t, x)$ directly, i.e. without stochastic simulations. It is well-known that $p(t, x)$ is the solution of the chemical master equation, but solving this equation numerically is a challenging task: since the solution has to be computed in each state of a huge state space, the number of degrees of freedom is far beyond the capacity of traditional methods. Novel methods for solving the chemical master equation have been constructed in [1, 8, 9, 10, 11, 12, 16, 17, 18, 20, 21, 23, 24, 25, 27, 30]. These methods are based on different approaches and assumptions, but they all have in common that the immense size of the problem is somehow reduced to a computationally manageable level. Generally speaking, the efficiency of each method depends mainly on its compression ratio, i.e. on the percentage of unknowns required to obtain the desired accuracy out of the total number of degrees of freedom.

The method advocated in this article is based on the representation of the solution in a sparse wavelet basis. In the wavelet basis, the number of *essential* degrees of freedom only amounts to a very small fraction of the total number of unknowns. This is due to the fact that the wavelet transform decomposes the input signal into information on a hierarchy of scales. Since smooth signals contain relatively few detail information, many coefficients of the wavelet representation nearly vanish and can be neglected if a tiny approximation error is accepted. Since the solution moves and changes as time evolves, however, the numerical method must not only propagate the coefficients of the essential basis elements, but also has to determine in each step *which* basis elements are currently the essential ones.

A prototype of such an adaptive wavelet method has been proposed in [20, 21] where a Galerkin ansatz with Rothe's method was combined with an iterative procedure that detects the essential degrees of freedom in each time step. Numerical experiments have shown the efficiency of this approach, but also revealed that two major improvements are possible:

- The wavelet used in [20, 21] was the Haar wavelet. The corresponding basis elements are particularly simple and allow efficient evaluations of the entries of the Galerkin matrix. The approximation in this basis, however, is usually far from optimal because the polynomial order of the Haar wavelet is only one. In the refined method presented here, the Haar basis is replaced by a higher-order basis of Daubechies wavelets. This yields a faster decay of the wavelet coefficients and hence a better compression ratio such that larger problems can be solved with a higher accuracy. We remark that other types of wavelets such as biorthogonal spline wavelets could also be used, and that this option is already available in our implementation of the method. In order to keep the presentation as simple as possible, however, only Daubechies wavelets are considered in this article.
- In [20, 21] the solution was propagated with a fixed step-size. This is inconvenient for two reasons. First, it is usually not clear *a priori* which step-size has to be chosen in order to obtain the desired accuracy. It would be clearly preferable to select the error tolerance and let the method choose the appropriate step-size by itself. Second, the choice of the step-size is often restricted by the stiff behavior of the solution in the initial phase whereas much larger time steps are possible later when $p(t, x)$ converges to a stationary distribution.

Hence, an adaptive step-size selection would yield important time savings for simulations. Such an adaptive time-stepping for the adaptive wavelet method is presented in Section 3.3. Our strategy is not based on comparison with an embedded method, but on analytical estimates for the errors caused by the approximations in time and space. Moreover, the second-order scheme used in [20, 21] is replaced by a fourth-order integrator.

In the next section we formulate the problem, introduce the chemical master equation and define our notation. In Section 3 the Haar wavelet method derived in [20, 21] is extended to higher-order Daubechies wavelets and provided with an adaptive step-size selection. Some parts of this section could be found in monographs on wavelet analysis or in [20, 21], respectively, but in order to make the exposition self-contained we have chosen to briefly compile the most important facts for the reader. Numerical examples are presented in Section 4. These tests demonstrate that with the new extensions – higher order wavelets for the space approximation and adaptive step-sizes for the time integration – the adaptive wavelet method is capable of solving large, non-trivial problems with bimodal solution profiles.

2. The chemical master equation

Suppose that the evolution of d species S_1, \dots, S_d interacting via K reaction channels is described by a Markov jump process on the state space \mathbb{N}^d . The entry $X_i(t) \in \mathbb{N}$ of a realization $X(t)$ is the number of particles of the i -th species at time t , where \mathbb{N} denotes the set of all nonnegative integers including zero. Our goal is to compute the probability distribution

$$p(t, x) = \mathbb{P}(X(t) = x), \quad x \in \mathbb{N}^d,$$

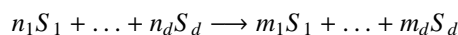
i.e. the probability that at time t there are exactly x_i particles of the i -th species ($i = 1, \dots, d$). It is well-known (see, e.g., [14]) that p evolves according to the *chemical master equation* (CME)

$$\begin{aligned} \partial_t p(t, \cdot) &= \mathcal{A}p(t, \cdot) \\ p(0, \cdot) &= p_0(\cdot) \end{aligned} \quad (1)$$

where \mathcal{A} denotes the operator

$$(\mathcal{A}p(t, \cdot))(x) = \sum_{k=1}^K (\alpha_k(x - \nu_k)p(t, x - \nu_k) - \alpha_k(x)p(t, x)) \quad (2)$$

and p_0 is a suitable probability distribution. We will occasionally omit the spatial variable and write, e.g., $p(t)$ instead of $p(t, x)$. In Equation (2), $\nu_k \in \mathbb{Z}^d$ is the stoichiometric vector which means that $X(t)$ jumps from the current state x to the new state $x + \nu_k$ if the k -th reaction channel fires. The term $\alpha_k(x) \geq 0$ is the propensity function of the k -th reaction channel. Roughly speaking, $\alpha_k(x)$ indicates how likely it is that the k -th reaction channel will fire in the next infinitesimal time interval (see [14] for details). Typically, the propensity function of the reaction



with $n_i, m_j \in \mathbb{N}$ and reaction constant $c > 0$ is given by

$$\alpha(x) = c \binom{x_1}{n_1} \dots \binom{x_d}{n_d}.$$

Reactions which can be inhibited are modelled with more complicated propensity functions; see Section 4 for examples. Since the term $x - \nu_k$ in (2) may have negative entries and the functions $p(t, x)$ and $\alpha_k(x)$ only make sense for integer nonnegative particle numbers $x = (x_1, \dots, x_d) \in \mathbb{N}^d$, we define $p(t, x) = 0$ and $\alpha_k(x) = 0$ for all $x \notin \mathbb{N}^d$.

Although the CME is actually defined on the infinite state space \mathbb{N}^d , numerical approximations are usually only computed on the truncated state space

$$\Omega_\xi = \{x \in \mathbb{N}^d : x_1 < \xi_1, \dots, x_d < \xi_d\}, \quad (3)$$

where $\xi = (\xi_1, \dots, \xi_d) \in \mathbb{N}^d$ is a suitably chosen truncation vector. It will always be assumed that ξ is so large that $p(t, \cdot)$ vanishes near the artificial border of Ω_ξ , such that the truncation error can be neglected. On the new boundaries, we impose the discrete Neumann boundary condition

$$\alpha_k(x) = 0 \quad \text{if } x \in \Omega_\xi \quad \text{and} \quad x + \nu_k \notin \Omega_\xi, \quad (4)$$

which suppresses all reactions leading from x to a state outside the truncated state space. This boundary condition guarantees that the solution of the truncated CME remains a probability distribution if p_0 is a probability distribution, that a stationary distribution exists, and that all nonzero eigenvalues have negative real part (cf., e.g., [20]). These favorable properties are lost if (4) is replaced by the discrete Dirichlet boundary condition

$$p(t, x) = 0 \quad \text{for all } x \in \mathbb{N}^d \setminus \Omega_\xi. \quad (5)$$

For small times, there is no visible difference between the two options if the probability distribution is located far away from the artificially imposed boundaries, but for longer time-intervals, the difference can be quite large. The reason is that Dirichlet boundary conditions imply

$$\sum_{x \in \Omega_\xi} \partial_t p(t, x) = \sum_{x \in \Omega_\xi} \mathcal{A}p(t, x) < 0.$$

Hence, the entire ‘‘probability mass’’ will leak out of the state space and as a consequence, the solution will converge to zero, albeit very slowly. Therefore, we use the discrete Neumann boundary condition (4) instead of (5).

The truncated state space is finite yet still very large in most applications (see Section 4 for examples). Since the solution $p(t, x)$ has to be computed for every state $x \in \Omega_\xi$, applying traditional methods to solve the CME is out of question for all but very small systems. The method introduced in [20, 21] compresses the huge number of degrees of freedom to a computationally tractable size by representing and propagating the solution in a sparse Haar wavelet basis. Numerical examples have shown the potential of the method, but also indicated that two extensions are necessary to handle larger problems: First, the Haar wavelet basis must be replaced by higher-order wavelets with better approximation properties, and second, the time integration must be enhanced by an adaptive time-stepping strategy. In this paper, we show how these two extensions can be integrated into the wavelet method from [20, 21].

3. Adaptive wavelet method

3.1. Daubechies’ orthogonal wavelets

The spatial approximation of the CME is not restricted to one particular class of wavelets. In the current implementation of our method, the user can choose between biorthogonal spline wavelets

and Daubechies wavelets, but for the sake of simplicity, only the case of Daubechies wavelets is discussed in this article.

Providing an elaborated introduction to Daubechies wavelets and their analysis is far beyond the scope of this paper. Expert knowledge of wavelet theory, however, is *not* required to understand how our method works, because we will only make use of certain *properties* of wavelets, not of the precise definitions and derivations. The purpose of this subsection is to compile the most important of these properties. For details, the reader is referred to the monographs [2, 7, 26].

For $j_0, j_{\max} \in \mathbb{N}$ the Daubechies wavelet basis of order $m \in \mathbb{N} \setminus \{0\}$ is a set of functions

$$\left\{ \varphi_{j_0,k}^{(m)} \mid k \in \mathbb{Z} \right\} \cup \left\{ \psi_{j,k}^{(m)} \mid k \in \mathbb{Z}, j = j_0, \dots, j_{\max} - 1 \right\} \subset L^2(\mathbb{R}) \quad (6)$$

with the following properties:

1. **Scaling function and mother wavelet.** There is a scaling function $\varphi^{(m)} \in L^2(\mathbb{R})$ and a mother wavelet $\psi^{(m)} \in L^2(\mathbb{R})$ such that all basis elements are obtained from these functions by shifts and dilations:

$$\begin{aligned} \varphi_{j_0,k}^{(m)}(x) &= 2^{j_0/2} \varphi^{(m)}(2^{j_0}x - k), \quad k \in \mathbb{Z} \\ \psi_{j,k}^{(m)}(x) &= 2^{j/2} \psi^{(m)}(2^jx - k), \quad j = j_0, \dots, j_{\max} - 1. \end{aligned}$$

The support of the scaling function and of the mother wavelet lies within a compact interval.

2. **Orthonormality.** For all $i, j \in \{j_0, \dots, j_{\max} - 1\}$ and $k, l \in \mathbb{Z}$, we have

$$\left\langle \varphi_{j_0,k}^{(m)}, \varphi_{j_0,l}^{(m)} \right\rangle_{L^2} = \delta_{k,l}, \quad \left\langle \psi_{j,k}^{(m)}, \psi_{i,l}^{(m)} \right\rangle_{L^2} = \delta_{j,i} \delta_{k,l}, \quad \left\langle \varphi_{j_0,k}^{(m)}, \psi_{i,l}^{(m)} \right\rangle_{L^2} = 0,$$

where $\delta_{k,l}$ is the Kronecker symbol.

3. **Refinement equations.** The scaling function $\varphi^{(m)}$ and the mother wavelet $\psi^{(m)}$ satisfy

$$\begin{aligned} \varphi^{(m)}(x) &= \sum_{n=0}^{2m-1} h_n \varphi^{(m)}(2x - n) \\ \psi^{(m)}(x) &= \sum_{n=2-2m}^1 g_n \varphi^{(m)}(2x - n) \end{aligned}$$

for certain filter coefficients $h_n \in \mathbb{R}$ and $g_n = (-1)^n h_{1-n}$. There is no explicit formula for the scaling function or the mother wavelet if $m > 1$. Values of $\varphi^{(m)}(x)$ and $\psi^{(m)}(x)$ can be computed with the *cascade algorithm* which, starting from the known values in certain points, computes intermediate values by applying the refinement equations iteratively. However, evaluations of $\varphi^{(m)}(x)$, $\psi^{(m)}(x)$ or the basis elements are rarely necessary. For the fast wavelet transform and its inverse (see property 4) only the filter coefficients h_0, \dots, h_{2m-1} are required, and these values can be looked up in the literature (see [26]).

4. **Fast wavelet transform.** Let $V_{j_{\max}}$ be the closure of the span of the wavelet basis (6), and let

$$f = \sum_{k \in \mathbb{Z}} a_{j_0,k}^{(m)} \varphi_{j_0,k}^{(m)} + \sum_{j=j_0}^{j_{\max}-1} \sum_{k \in \mathbb{Z}} b_{j,k}^{(m)} \psi_{j,k}^{(m)} \quad (7)$$

be the representation of a function $f \in V_{j_{\max}}$. Since numerical computations can only be performed on a bounded interval, we assume that the support of f is finite. Hence, only a finite number of coefficients $a_{j_0,k}^{(m)} = \langle \varphi_{j_0,k}^{(m)}, f \rangle_{L^2}$ and $b_{j,k}^{(m)} = \langle \psi_{j,k}^{(m)}, f \rangle_{L^2}$ are nonzero. These coefficients can be very efficiently computed by the fast wavelet transform, which proceeds recursively and exploits the properties 1 and 3. Conversely, reconstructing a function from its coefficients can be efficiently accomplished by the inverse fast wavelet transform.

5. The scaling functions and the mother wavelet generate a **hierarchy of spaces** $V_{j_0} \subset \dots \subset V_{j_{\max}} \subset L^2(\mathbb{R})$ defined by

$$V_{j+1} = V_j \oplus W_j = V_{j_0} \oplus W_{j_0} \oplus W_{j_0+1} \oplus \dots \oplus W_j$$

and

$$V_{j_0} = \overline{\text{span}\{\varphi_{j_0,k}^{(m)} \mid k \in \mathbb{Z}\}}, \quad W_j = \overline{\text{span}\{\psi_{j,k}^{(m)} \mid k \in \mathbb{Z}\}}.$$

Let $P_i : L^2(\mathbb{R}) \rightarrow V_i$ denote the orthogonal projector onto V_i . Note that $P_i f$ is obtained by setting $b_{j,k}^{(m)} = 0$ for all $j \geq i$ in (7). Then, every $f \in V_{j_{\max}}$ can be represented as

$$f = P_{j_{\max}} f = P_{j_0} f + \sum_{j=j_0}^{j_{\max}-1} (P_{j+1} - P_j) f.$$

The term

$$P_{j_0} f = \sum_{k \in \mathbb{Z}} a_{j_0,k}^{(m)} \varphi_{j_0,k}^{(m)} \in V_{j_0}$$

approximates the function on the coarsest scale, whereas the terms

$$(P_{j+1} - P_j) f = \sum_{k \in \mathbb{Z}} b_{j,k}^{(m)} \psi_{j,k}^{(m)} \in W_j$$

represent the new detail information which can be captured when the space V_j is enlarged to $V_{j+1} = V_j \oplus W_j$. If $P_{j_{\max}} f$ is sufficiently smooth, one would expect that most of the information is already observable on the coarsest scales, and that for large j the coefficients $b_{j,k}^{(m)}$ corresponding to the detail information $(P_{j+1} - P_j) f$ are close to zero. Hence, the wavelet basis allows to *compress* the data: if a small approximation error is acceptable, then only the few non-vanishing coefficients have to be stored instead of the entire representation (7). A mathematically more precise formulation of this fact is stated in property 7.

6. **Vanishing moments.** The mother wavelet $\psi^{(m)}(x)$ of order m satisfies

$$\int_{\mathbb{R}} x^n \psi^{(m)}(x) dx = 0 \quad \text{for all } n = 0, \dots, m-1.$$

This means that all polynomials with a degree less than m are contained in V_{j_0} .

7. **Wavelet compression.** Let $f \in C_0^n(\mathbb{R})$ for some $n \in \{1, \dots, m\}$. Then, for all $j \in \{j_0, \dots, j_{\max}\}$, the projection error is bounded by

$$\|f - P_j f\|_{L^2} \leq C 2^{-nj} \|f^{(n)}(x)\|_{L^2}. \quad (8)$$

This estimate confirms that omitting details by discarding the corresponding coefficients can still yield a very precise approximation. Note that the approximation error does not only depend on the smoothness of the function and on the truncation level j , but also on the order m of the wavelet, because (8) only holds for $n \leq m$. Similar estimates can be shown under weaker regularity assumptions, cf. [2].

Example: Haar wavelets. The Daubechies wavelet with $m = 1$ is the Haar wavelet. The scaling function is $\varphi^{(m)}(x) = \chi_{[0,1)}(x)$, and the mother wavelet is $\psi(x) = \chi_{[0,1/2)}(x) - \chi_{[1/2,1)}(x)$, with χ_I denoting the characteristic function of the interval I . The filter coefficients of the refinement equations are $h_0 = h_1 = g_0 = 1$ and $g_1 = -1$. The space V_j contains all functions which are constant on all intervals $2^{-j}k + [0, 2^{-j})$, $k \in \mathbb{Z}$. For increasing j , the projection $P_j f$ captures more and more details of the input function f . However, only polynomials of degree $m - 1 = 0$ can be represented exactly, and the estimate (8) is only true for $n = 1$. Hence, the approximation properties of the Haar wavelet basis are rather modest. This was our motivation to use higher-order Daubechies wavelets instead of the Haar wavelet to approximate the solution of the CME.

Example: Daubechies db2 wavelets. The scaling function and the mother wavelet for $m = 2$ are shown in Figure 1. There is no explicit formula for these functions. Both functions are zero outside the interval $[0, 3]$. The filter coefficients are

$$h_0 = \frac{1 + \sqrt{3}}{4\sqrt{2}}, \quad h_1 = \frac{3 + \sqrt{3}}{4\sqrt{2}}, \quad h_2 = \frac{3 - \sqrt{3}}{4\sqrt{2}}, \quad h_3 = \frac{1 - \sqrt{3}}{4\sqrt{2}} \quad \text{and} \quad g_n = (-1)^n h_{1-n}.$$

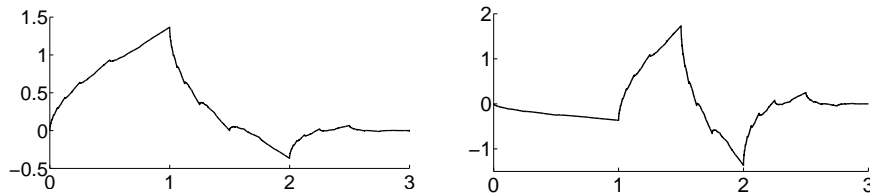


Figure 1: Scaling function (left) and mother wavelet (right) of the db2 wavelet ($m = 2$).

Wavelets on Ω_ξ . The elements of the wavelet basis defined above are functions on \mathbb{R} . For application to the CME, however, we need a wavelet basis of functions on the bounded, discrete and multi-dimensional domain $\Omega_\xi \subset \mathbb{N}^d$. We briefly sketch how this adaptation can be made.

There are several ways to define a wavelet basis in more than one spatial dimension. A straightforward option is to use tensor products of one-dimensional basis elements, but this construction does not generate a *multiresolution analysis* (cf. Section 2.2 in [2]). An alternative is described in Sections 1.4 and 2.12 of [2]. In our method both options can be used, and both options were implemented in our MATLAB code.

A wavelet representation of a function on a bounded interval can be obtained by first extending the target function to \mathbb{R} by periodic continuation and then applying the above setting. The disadvantage of this strategy is that the extension will introduce an artificial discontinuity at the boundaries, which will unnecessarily increase the number of wavelet coefficients required to represent the target function. This inconvenience can be avoided with special wavelets designed for bounded intervals; cf. Section 2.12 in [2]. Our method is compatible with each of these alternatives, but for simplicity, periodic continuation is used in the current implementation of our code.

As soon as wavelets on a bounded interval $[a, b]$ have been defined, one can introduce an equidistant grid $x_n = a + n(b - a)/2^r$ with $r \in \mathbb{N}$ and $n = 0, \dots, 2^r - 1$. Then, every function f on that grid can be identified with a function on the discrete state space $\{0, \dots, 2^r - 1\}$ via $\tilde{f}(n) = f(x_n)$. In the multivariate case, this defines a wavelet basis for functions on Ω_ξ .

Summary. Let $\mathcal{H}(\Omega_\xi)$ be the linear space of all discrete functions $f : \Omega_\xi \rightarrow \mathbb{R}$. On this space, we define the inner product

$$\langle f, g \rangle = \sum_{x \in \Omega_\xi} f(x)g(x), \quad f, g \in \mathcal{H}(\Omega_\xi), \quad (9)$$

and the norm $\|f\|_2 = \sqrt{\langle f, f \rangle}$. Let $N = \xi_1 \cdot \dots \cdot \xi_d$ be the total number of states. With the construction sketched in this subsection, one obtains an orthonormal, discrete wavelet basis $\{v_1^{(m)}, \dots, v_N^{(m)}\}$ of $\mathcal{H}(\Omega_\xi)$. For simplicity, the basis elements are enumerated by one single index instead of (multi-)indices for level and number. The properties of this basis are inherited from the univariate, continuous setting. The coefficients $a_j^{(m)} = \langle f, v_j^{(m)} \rangle$ in the representation $f = \sum_{j=1}^N a_j^{(m)} v_j^{(m)}$ can be computed with $O(N)$ operations with the fast wavelet transform. It is useful to remark that because we operate in a discrete setting, computing the finest scaling coefficients presents no problems, even though the Daubechies scaling functions have no explicit representation, as these are taken to be the values of the function f itself. Conversely, the function f can be reconstructed from the coefficients $a_j^{(m)}$ with $O(N)$ operations via the fast inverse wavelet transform. If f is sufficiently regular, then a higher order m implies a faster decay of the coefficients and thus a better compression rate when vanishing coefficients are discarded. As a consequence of the orthonormality, the compression error in the norm $\|\cdot\|_2$ is the same as the 2-norm error of the discarded coefficients.

The basis elements are somewhat abstract because there are no explicit formulas for the v_j . As in the univariate, continuous setting, these functions are defined recursively via the discrete counterpart of the refinement equations. The fast wavelet transform and its inverse do not use the basis elements explicitly. These transforms can be thought of as black boxes which, given either the input function or the coefficients of the wavelet representation, return the corresponding counterpart. In the adaptive wavelet method constructed in the following subsections, the only part where the basis elements have to be evaluated is the computation of the Galerkin matrix (17).

3.2. Approximation with fixed step-size

In this subsection we show how the solution of the CME (1) can be approximated adaptively in the wavelet basis. As a first step, we consider a *fixed* time step $h > 0$ and concentrate on the question how the essential degrees of freedom can be detected and propagated. The algorithm described below is essentially the one proposed in [20, 21]. The difference is that the latter used only a second-order scheme for the time integration and the Haar basis for the spatial approximation. A strategy for adaptively selecting the step-size is presented in Subsection 3.3.

Let $\{v_1^{(m)}, \dots, v_N^{(m)}\}$ be the orthonormal, discrete wavelet basis from the previous subsection. It is assumed that the polynomial order m of the wavelet is chosen by the user, and the index “(m)” will from now on be omitted. Let

$$p_n = \sum_{i=1}^{\eta} \beta_i v_{j_i} \approx p(t_n) \quad (10)$$

be the numerical approximation available at time $t_n = t_0 + nh$. Here, $\{j_1, \dots, j_\eta\}$ is a small subset of the index set $\{1, \dots, N\}$, and $\beta = (\beta_1, \dots, \beta_\eta)^T \in \mathbb{R}^\eta$ is the coefficient vector of p_n . The function p_n is supposed to be propagated by one step of the 2-stage Gauss-Runge-Kutta method. For linear problems this method is equivalent to the (2,2)-Padé approximation to the

exponential function, and its order (order 4) is the highest possible among all integrators with two stages. Moreover, the method is A-stable, which is important because the real parts of all eigenvalues of the operator \mathcal{A} are nonpositive and the CME can be very stiff in the initial phase. Performing one time step means that the new approximation $u_{n+1} \approx p(t_{n+1})$ must be computed by solving the linear equation

$$Q(h\mathcal{A})u_{n+1} = P(h\mathcal{A})p_n \quad (11)$$

with

$$Q(h\mathcal{A}) = I - \frac{h}{2}\mathcal{A} + \frac{h^2}{12}\mathcal{A}^2, \quad P(h\mathcal{A}) = I + \frac{h}{2}\mathcal{A} + \frac{h^2}{12}\mathcal{A}^2. \quad (12)$$

Here and below, I denotes the identity operator/matrix. An equivalent formulation is

$$u_{n+1} = p_n + \frac{h}{2}(g_1 + g_2) \quad (13)$$

where (g_1, g_2) is the solution of

$$\begin{pmatrix} I - \frac{h}{4}\mathcal{A} & -h\left(\frac{1}{4} - \frac{\sqrt{3}}{6}\right)\mathcal{A} \\ -h\left(\frac{1}{4} + \frac{\sqrt{3}}{6}\right)\mathcal{A} & I - \frac{h}{4}\mathcal{A} \end{pmatrix} \begin{pmatrix} g_1 \\ g_2 \end{pmatrix} = \begin{pmatrix} \mathcal{A}p_n \\ \mathcal{A}p_n \end{pmatrix}. \quad (14)$$

Unfortunately, neither (11) nor (14) can be solved in a straightforward way because both linear systems are far too large for standard direct or iterative schemes. The *exact* solution, however, is not required – it is sufficient to approximate u_{n+1} up to an error which does not significantly increase the local error of the time integration. According to the properties of the wavelet basis it can be expected that an approximation $p_{n+1} \approx u_{n+1}$ can be found in a low-dimensional subspace of $\mathcal{H}(\Omega_\xi)$. A first candidate for this subspace is the span of $\{v_{j_1}, \dots, v_{j_\eta}\}$, i.e. the subspace of the previous step. An approximation

$$p_{n+1}^{(0)} = \sum_{i=1}^{\eta} \gamma_i^{(0)} v_{j_i}, \quad p_{n+1}^{(0)} = p_n + \frac{h}{2}(g_1^{(0)} + g_2^{(0)}), \quad g_s^{(0)} = \sum_{i=1}^{\eta} \zeta_{s,i}^{(0)} v_{j_i}, \quad s \in \{1, 2\} \quad (15)$$

in this space is obtained by imposing the Galerkin conditions

$$\begin{aligned} \langle v_{j_i}, (I - \frac{h}{4}\mathcal{A})g_1^{(0)} \rangle - h\left(\frac{1}{4} - \frac{\sqrt{3}}{6}\right)\langle v_{j_i}, \mathcal{A}g_2^{(0)} \rangle &= \langle v_{j_i}, \mathcal{A}p_n \rangle \\ -h\left(\frac{1}{4} + \frac{\sqrt{3}}{6}\right)\langle v_{j_i}, \mathcal{A}g_1^{(0)} \rangle + \langle v_{j_i}, (I - \frac{h}{4}\mathcal{A})g_2^{(0)} \rangle &= \langle v_{j_i}, \mathcal{A}p_n \rangle \end{aligned} \quad (16)$$

for all $i = 1, \dots, \eta$. Let $A \in \mathbb{R}^{\eta \times \eta}$ denote the Galerkin matrix defined by

$$A = (a_{ik})_{i,k=1}^{\eta}, \quad a_{ik} = \langle v_{j_i}, \mathcal{A}v_{j_k} \rangle. \quad (17)$$

Then, (16) can be rewritten as

$$\begin{pmatrix} I - \frac{h}{4}A & -h\left(\frac{1}{4} - \frac{\sqrt{3}}{6}\right)A \\ -h\left(\frac{1}{4} + \frac{\sqrt{3}}{6}\right)A & I - \frac{h}{4}A \end{pmatrix} \begin{pmatrix} \zeta_1^{(0)} \\ \zeta_2^{(0)} \end{pmatrix} = \begin{pmatrix} A\beta \\ A\beta \end{pmatrix} \quad (18)$$

where $\zeta_s^{(0)} = (\zeta_{s,1}^{(0)}, \dots, \zeta_{s,\eta}^{(0)})^T$. Since the Galerkin matrix $A \in \mathbb{R}^{\eta \times \eta}$ is much smaller than $\mathcal{A} \in \mathbb{R}^{N \times N}$, (18) can be solved with GMRES or other iterative methods. The approximation $p_{n+1}^{(0)}$ is then obtained by a fast inverse wavelet transform of the new coefficient vector $\gamma^{(0)} = \beta + \frac{h}{2}(\zeta_1^{(0)} + \zeta_2^{(0)})$. Due to the equivalence of (11) and (14), $\gamma^{(0)}$ solves the equation

$$Q(hA)\gamma^{(0)} = P(hA)\beta. \quad (19)$$

Of course, $p_{n+1}^{(0)}$ does in general *not* coincide with the solution u_{n+1} of the full problem (11). Let

$$r^{(0)} = Q(h\mathcal{A})p_{n+1}^{(0)} - P(h\mathcal{A})p_n$$

be the residual. The Galerkin condition (16) implies that the residual is orthogonal to the approximation space. Hence, the residual shows which “part” of the full problem has been neglected by solving (19) instead of (11). Roughly speaking, the value $\langle v_k, r^{(0)} \rangle$ (i.e. the coefficient of the residual in the wavelet basis) tells “how much the residual points into the direction of v_k ”. If $|\langle v_k, r^{(0)} \rangle|$ is large, then the approximation will probably improve if v_k is added to the current basis. Note that $\langle v_k, r^{(0)} \rangle = 0$ if v_k is already contained in the selection of basis elements.

Now the adaptive wavelet method proceeds as follows. First, the basis is enlarged by a fixed number $\Delta\mu$ of new elements. The new elements $v_{j_{\eta+1}}, \dots, v_{j_{\eta+\Delta\mu}}$ are those which yield the largest values $|\langle v_k, r^{(0)} \rangle|$. Next, the Galerkin matrix (17) is updated by adding $\Delta\mu$ new lines and columns corresponding to $v_{j_{\eta+1}}, \dots, v_{j_{\eta+\Delta\mu}}$. Then, solving the system (19) with the enlarged A yields a refined coefficient vector $\gamma^{(1)}$ and an improved approximation $p_{n+1}^{(1)} = \sum_{i=1}^{\eta^{(1)}} \gamma_i^{(1)} v_{j_i}$ with $\eta^{(1)} = \eta + \Delta\mu$ terms. Iterating this procedure leads to a sequence of approximations $p_{n+1}^{(0)}, p_{n+1}^{(1)}, p_{n+1}^{(2)}, \dots$ in a hierarchy of increasing approximation spaces. As soon as $\|r^{(\ell)}\|$ is smaller than the chosen tolerance, the iteration is stopped, and the approximation $p_{n+1}^{(\ell)}$ is accepted.

In order to prevent unlimited growth of the number of basis elements, all dispensable terms are removed from the representation of $p_{n+1}^{(\ell)} = \sum_{i=1}^{\eta^{(\ell)}} \gamma_i^{(\ell)} v_{j_i}$ in a post-processing step. Let $\mathcal{I} \subset \{1, \dots, \eta^{(\ell)}\}$ be a subset of the index set, and let $p_{n+1}^{[\mathcal{I}]} = \sum_{i \in \mathcal{I}} \gamma_i^{(\ell)} v_{j_i}$ be the approximation obtained by deleting all terms with $i \notin \mathcal{I}$ from the representation. Since the wavelet basis is orthonormal, the truncation error in $\|\cdot\|_2$ is the 2-norm of the discarded coefficients, i.e.

$$\|p_{n+1}^{(\ell)} - p_{n+1}^{[\mathcal{I}]}\|_2 = \sqrt{\sum_{i \notin \mathcal{I}} (\gamma_i^{(\ell)})^2}.$$

In order to reach an accuracy $\|p_{n+1}^{(\ell)} - p_{n+1}^{[\mathcal{I}]}\|_2 \leq \text{tol}_{trunc}$ with a minimal number of basis elements, we simply order the coefficients by magnitude and truncate the smallest coefficients as long as $\sum_{i \notin \mathcal{I}} (\gamma_i^{(\ell)})^2 \leq \text{tol}_{trunc}^2$. If the error is measured with respect to $\|\cdot\|_1$ rather than $\|\cdot\|_2$, we choose $\text{tol}_{trunc} \leq c \cdot \text{tol}$ where c is a factor which accounts for the equivalence of the norms. The choice $c = 1/\sqrt{N}$ is correct, but often too pessimistic. A more optimistic choice can be made by replacing N by the number of states where $p_{n+1}^{(\ell)}$ is essentially larger than zero. The function $p_{n+1} := p_{n+1}^{[\mathcal{I}]}$ obtained by thresholding is the final result of the entire time step.

The following algorithm sketches one single time step of the adaptive wavelet method with fixed step size. The algorithm does not store $p_{n+1}^{(0)}, p_{n+1}^{(1)}, p_{n+1}^{(2)}, \dots$ but only one single function \hat{p}_{n+1} which is overwritten in each iteration.

Parameter: step-size $h > 0$, tolerance tol , safety factor C_r (see remark 3) below)

Input: index subset $\{j_1, \dots, j_\eta\}$ and coefficients $\beta_1, \dots, \beta_\eta$ of the current approximation $p_n = \sum_{i=1}^\eta \beta_i v_{j_i}$
Galerkin matrix A defined by (17)

Output: index subset $\{k_1, \dots, k_\mu\}$ and coefficients $\gamma_1, \dots, \gamma_\mu$ of the new approximation $p_{n+1} = \sum_{i=1}^\mu \gamma_i v_{k_i}$
updated Galerkin matrix

1. Set $\hat{\mu} = \eta$.
2. Solve the linear system

$$\begin{pmatrix} I - \frac{h}{4}A & -h\left(\frac{1}{4} - \frac{\sqrt{3}}{6}\right)A \\ -h\left(\frac{1}{4} + \frac{\sqrt{3}}{6}\right)A & I - \frac{h}{4}A \end{pmatrix} \begin{pmatrix} \zeta_1 \\ \zeta_2 \end{pmatrix} = \begin{pmatrix} A\hat{\beta} \\ A\hat{\beta} \end{pmatrix}$$

and set $\hat{\gamma} = \hat{\beta} + \frac{h}{2}(\zeta_1 + \zeta_2)$. The vector $\hat{\beta}$ is an embedding of $\beta \in \mathbb{R}^\eta$ into $\mathbb{R}^{\hat{\mu}}$:

$$\hat{\beta} = (\beta_1, \dots, \beta_\eta, \underbrace{0, \dots, 0}_{\hat{\mu}-\eta})^T \quad (20)$$

3. Compute the new approximation $\hat{p}_{n+1} = \sum_{i=1}^{\hat{\mu}} \hat{\gamma}_i v_{j_i}$ by a fast inverse wavelet transform.
4. Compute the residual $r = Q(h\mathcal{A})\hat{p}_{n+1} - P(h\mathcal{A})p_n$ with Q and P defined by (12).
5. If $\|r\|_1 > C_r \cdot \text{tol}$:
 - (a) Compute $\chi_l = |\langle v_l, r \rangle|$ for $l = 1, \dots, N$ by a fast wavelet transform.
 - (b) Find the indices $j_{\hat{\mu}+1}, \dots, j_{\hat{\mu}+\Delta\mu}$ of the $\Delta\mu$ largest entries of (χ_1, \dots, χ_N) .
 - (c) Add $v_{j_{\hat{\mu}+1}}, \dots, v_{j_{\hat{\mu}+\Delta\mu}}$ to the current selection of basis elements.
 - (d) Update the Galerkin matrix by adding new blocks corresponding to the new basis vectors:

$$A = (a_{ik})_{i,k=1}^{\hat{\mu}+\Delta\mu}, \quad a_{ik} = \langle v_{j_i}, \mathcal{A}v_{j_k} \rangle.$$

- (e) Set $\hat{\mu} \mapsto \hat{\mu} + \Delta\mu$.
- (f) Go to step 2.

6. The result $p_{n+1} = \sum_{i=1}^\mu \gamma_i v_{k_i}$ is obtained by discarding all coefficients γ_i with $i \notin \mathcal{I}$, where \mathcal{I} is the index set of the largest coefficients. The number of coefficients is chosen in such a way that $\|p_{n+1} - \hat{p}_{n+1}\|_1 \leq \text{tol}$ (see above). The corresponding columns and lines are deleted from the Galerkin matrix A .

Remarks.

1. The adaptive wavelet method described here is closely related to similar methods for solving elliptic and parabolic partial differential equations; cf. [3, 4, 5, 6, 28, 29]
2. It is not advisable to compute the coefficient vector by solving (19) because the matrix A^2 which occurs in $Q(hA)$ typically increases the condition number tremendously. This is avoided in the equivalent formulation (18). The price to pay, however, is the fact that the linear system (18) has twice as many unknowns as (19). The doubling of the linear system

can be avoided if the 2-stage Gauss-Runge-Kutta method is replaced by a singly diagonally implicit Runge-Kutta method (cf. Section IV.6 in [15]) where only linear systems with the same matrix $(I - chA) \in \mathbb{R}^{7 \times 7}$, but different right-hand sides have to be solved. Unfortunately, singly diagonally implicit Runge-Kutta method are either less accurate (order 3 with two stages) or require more stages (three stages for order 4) than the 2-stage Gauss-Runge-Kutta method. Therefore, it is difficult to say *a priori* which method will be more efficient for a particular problem. We have tested both possibilities on several applications and did not notice a significant difference in efficiency, as the computationally critical part of the algorithm is the assembly of the Galerkin matrix A in Step 5(d) of the algorithm described above.

3. The iteration terminates if $\|r\|_1 > C_r \cdot \text{tol}$. The safety factor $C_r \leq 1$ in step 5 can be chosen as follows. If $\|r\|_1 \leq C_r \cdot \text{tol}$, then comparing $Q(h\mathcal{A})u_{n+1} = P(h\mathcal{A})p_n$ (cf. (11)) and $Q(h\mathcal{A})\hat{p}_{n+1} = P(h\mathcal{A})p_n + r$ yields the error bound

$$\|u_{n+1} - \hat{p}_{n+1}\|_1 \leq \|Q(h\mathcal{A})^{-1}r\|_1 \leq \|Q(h\mathcal{A})^{-1}\|_1 \cdot C_r \cdot \text{tol}. \quad (21)$$

In order to conclude that $\|u_{n+1} - \hat{p}_{n+1}\|_1 \leq \text{tol}$, we have to choose $C_r = 1/\|Q(h\mathcal{A})^{-1}\|_1$. Based on our numerical experiments, we conjecture that $\|Q(h\mathcal{A})^{-1}\|_1 = 1$, but unfortunately we were not able to prove this. However, since $(I - h\mathcal{A}/2)^{-1}$ is known to be contractive and $Q(h\mathcal{A})^{-1}$ is a higher order perturbation, choosing $C_r \lesssim 1$ seems to be reasonable.

4. The limited memory of the computer imposes an upper bound for the maximal number of used basis elements. Thus, it is sometimes more convenient to prescribe the number of degrees of freedom instead of the accuracy of the approximation. In this case, the number of basis elements which are kept *after* the time-step (μ) is chosen by the user. Moreover, one can select a second parameter μ_{max} which denotes the maximal number of basis elements *during* the time-step. The condition “If $\|r\|_1 > C_r \cdot \text{tol}$ ” in step 5 is then replaced by “If $\|r\|_1 > C_r \cdot \text{tol}$ and $\hat{\mu} + \Delta\mu \leq \mu_{max}$ ”.
5. If the time step is rather large, then propagating the approximation sometimes demands much more basis elements than representing it. In such a situation, many of the basis elements discarded at the end of a time step are selected again in the next time step. This decreases the efficiency of the algorithm, because the corresponding entries in the Galerkin matrix have to be computed once again. It is thus advantageous to fix a *lower* bound μ_{min} for the number of degrees of freedom in step 6.
6. Since p_{n+1} is supposed to approximate a probability distribution, special care must be taken to ensure positivity of the numerical solution. This issue is not only related to the space approximation, where the oscillatory nature of the wavelet basis combined with coefficient thresholding can potentially lead to negative values for the approximated function, but also affects time integration. Even if the spatial representation would be exact, meaning that all the basis elements are used to approximate the function, the numerical solution could still have negative entries if rather large-steps are used. This is because most Runge-Kutta methods only respect positivity if the time-step is sufficiently small (an exception is the implicit Euler method applied to $\dot{y} = -My$ where M is a M -matrix; in this case, the numerical solution is positive for every step-size $h > 0$ if $y_0 \geq 0$). Hence, methods based on other spatial approximations do not always preserve positivity, either. However, the knowledge that the exact solution must be nonnegative can be used as an indicator. If negative entries appear in the probability distribution, then the step-size must be decreased

and/or the number of basis elements must be increased. We remark that in our numerical tests loss of positivity only happened when problems with a stiff transient phase were approximated with very low tolerances.

3.3. Adaptive step-size control

Up to now, the wavelet method was adaptive in space, but not in time. Solving chemical master equations with a fixed step-size, however, can be rather inefficient because often the short stiff transient phase at the beginning of the time interval imposes severe step-size restrictions whereas much larger time steps can be made towards the end.

In this subsection we introduce a strategy to select the step-size adaptively¹ in such a way that the (local) approximation error remains under or close to the chosen tolerance τ_{ol} . Strictly speaking, the error bounds given below will only guarantee that the error is smaller than $C \cdot \tau_{\text{ol}}$ with some (moderate) constant $C > 1$. If it is of crucial importance to keep the error always below the tolerance, this can be achieved by introducing an appropriate safety factor.

We start with the following bound for the local error.

Theorem 1. *Let p_n be the approximation computed in the n -th time step with tolerance $\tau_{\text{ol}} > 0$ and let $p(t)$ be the exact solution of the CME*

$$\begin{aligned} \dot{p}(t) &= \mathcal{A}p(t) \quad \text{for } t \in [t_n, t_{n+1}] \\ p(t_n) &= p_n \end{aligned} \quad (22)$$

which starts from p_n at time t_n . Suppose that the representation of \hat{p}_{n+1} before the truncation (step 6) is $\hat{p}_{n+1} = \sum_{i=1}^{\hat{\mu}} \hat{\gamma}_i v_{j_i}$ and let

$$V = \text{span}\{v_{j_1}, \dots, v_{j_{\hat{\mu}}}\} \subset \mathcal{H}(\Omega_\xi)$$

be the iteratively enlarged approximation space. Note that $p_n \in V$ because V is the approximation space **before** the truncation step. Let $q(t)$ be the solution of the projected CME

$$\begin{aligned} \dot{q}(t) &= P_V \mathcal{A}q(t) \quad \text{for } t \in [t_n, t_{n+1}] \\ q(t_n) &= p_n \end{aligned} \quad (23)$$

where

$$P_V : \mathcal{H}(\Omega_\xi) \longrightarrow V, \quad P_V w = \sum_{i=1}^{\hat{\mu}} \langle v_{j_i}, w \rangle v_{j_i}$$

denotes the orthogonal projection from $\mathcal{H}(\Omega_\xi)$ onto V . Then, the local error $p_{n+1} - p(t_{n+1})$ is bounded by

$$\|p_{n+1} - p(t_{n+1})\|_1 \leq \frac{h^5}{720} \|(P_V \mathcal{A})^5 p_n\|_1 + \mathcal{O}(h^6) + \tau_{\text{ol}} + \int_{t_n}^{t_{n+1}} \|(P_V - I) \mathcal{A}q(s)\|_1 ds. \quad (24)$$

¹We remark that *adaptive* in this context means the ability to control the global step-size and not a fully adaptive scheme where each degree of freedom is propagated with its own step-size. Although such a local time stepping method could increase performance, it is dependent on advanced knowledge of the problem, and our goal is to construct a method that is free of such assumptions.

Proof. The error is split into the three parts

$$\|p_{n+1} - p(t_{n+1})\|_1 \leq \|p_{n+1} - \hat{p}_{n+1}\|_1 + \|\hat{p}_{n+1} - q(t_{n+1})\|_1 + \|q(t_{n+1}) - p(t_{n+1})\|_1. \quad (25)$$

The error bound $\|p_{n+1} - \hat{p}_{n+1}\|_1 \leq \tau_0 1$ follows directly from the definition of p_{n+1} in step 6 of the algorithm. The steps 2 and 3 in the algorithm are equivalent to applying the 2-stage Gauss method to the projected CME (23). The local error of the Gauss method is bounded by

$$\|\hat{p}_{n+1} - q(t_{n+1})\| \leq \frac{h^5}{720} \|(P_V \mathcal{A})^5 p_n\|_1 + \mathcal{O}(h^6) \quad (26)$$

which can be shown by standard arguments. In order to derive an error bound for the last term in (25) we use that $d(t) = q(t) - p(t)$ satisfies the equation

$$\dot{d}(t) = \mathcal{A}d(t) + (P_V - I)\mathcal{A}q(t).$$

The variation-of-constants formula yields

$$d(t) = d(t_n) + \int_{t_n}^t \exp((t-s)\mathcal{A})(P_V - I)\mathcal{A}q(s)ds$$

where $\exp((t-t_n)\mathcal{A})$ denotes the flow of the CME (22). Since $d(t_n) = q(t_n) - p(t_n) = 0$ and $\|\exp((t-t_n)\mathcal{A})\|_1 = 1$ for all $t \geq t_n$, it follows that

$$\|q(t_{n+1}) - p(t_{n+1})\|_1 = \|d(t_{n+1})\|_1 \leq \int_{t_n}^{t_{n+1}} \|(P_V - I)\mathcal{A}q(s)\|_1 ds.$$

Substituting these bounds in (25) proves the assertion. \blacksquare

In (24) the term $h^5 \|(P_V \mathcal{A})^5 p_n\|_1 / 720$ arises from the time integration of the *projected* CME. Evaluating the expression $(P_V \mathcal{A})^5 p_n$ in a straightforward way would imply five evaluations of \mathcal{A} , but fortunately, this can easily be avoided: if $p_n = \sum_{i=1}^{\hat{\mu}} \beta_i v_{j_i}$ is the representation of the old approximation, then

$$(P_V \mathcal{A})^5 p_n = \sum_{i=1}^{\hat{\mu}} \zeta_i v_{j_i}, \quad (\zeta_1, \dots, \zeta_{\hat{\mu}})^T = A^5 (\beta_1, \dots, \beta_{\hat{\mu}})^T.$$

Hence, only the relatively small Galerkin matrix A has to be applied five times, not the full operator \mathcal{A} . The integral term in (24) represents the error caused by the spatial approximation in the sense that it describes how the solution of the *projected* CME deviates from the solution of the *full* CME. Since the function $q(t)$ is not computed in the algorithm, an exact evaluation of this term is not available, but a first-order approximation is given by

$$\int_{t_n}^{t_{n+1}} \|(P_V - I)\mathcal{A}q(s)\|_1 ds \approx (t_{n+1} - t_n) \|(P_V - I)\mathcal{A}q(t_n)\|_1 = h \|(P_V - I)\mathcal{A}p_n\|_1. \quad (27)$$

With (24) and (27) the condition $\|p(t_{n+1}) - p_{n+1}\|_1 \approx \text{tol}$ leads to the step-size selection

$$h = \min \left\{ \frac{\text{tol}}{\|(P_V - I)\mathcal{A}p_n\|_1}, C_{safe} \cdot \left(\frac{720 \cdot \text{tol}}{\|(P_V \mathcal{A})^5 p_n\|_1} \right)^{1/5} \right\} \quad (28)$$

with an optional safety factor $C_{safe} \leq 1$. The main difficulty is that the step-size h has to be chosen *before* the time step $p_n \mapsto p_{n+1}$ is carried out, but the space V is only known *after* the time step. At time t_n only the subspace

$$W = \text{span}\{v_{j_1}, \dots, v_{j_\eta}\} \subset V \subset \mathcal{H}(\Omega_\xi)$$

spanned by the basis elements from the representation $p_n = \sum_{i=1}^\eta \beta_i v_{j_i} \approx p(t_n)$ is available. For the estimate (26), this difference is negligible, because this term estimates the error caused by the time integration. For the estimate of the spatial error, however, simply replacing V by W is far too pessimistic. An estimate for the term $\|(I - P_V)\mathcal{A}p_n\|_1$ can be computed by means of a prediction of how many new basis elements will be chosen during the time step. Let

$$\mathcal{A}p_n = \sum_{l=1}^N \theta_l v_l \quad (29)$$

be the representation of $\mathcal{A}p_n$. First, we apply the projection $(I - P_W)$ which removes all terms with index $l \in \{j_1, \dots, j_\eta\}$ from (29). From the remaining coefficients, we discard the m coefficients with the largest absolute value, because the corresponding basis elements are most likely to be selected during the enlargement of the approximation space. The number m should depend on how many basis elements are currently used (η) and on the maximal number of basis elements (μ_{max}), i.e. $m = s \cdot (\mu_{max} - \eta)$ with some safety factor $s \in [0, 1]$. In our numerical experiments, the value $s = 0.5$ was used. The larger the value m , the larger the new step-size h , but this also means that more basis elements will be necessary.

These considerations are summarized in the following algorithm for the step-size selection. This algorithm is started at the beginning of every time step, i.e. before the algorithm from Subsection 3.2.

Parameter: error tolerance $\text{tol} > 0$

Input: index subset $\{j_1, \dots, j_\eta\}$ and coefficients $\beta_1, \dots, \beta_\eta$ of the current approximation $p_n = \sum_{i=1}^\eta \beta_i v_{j_i}$
Galerkin matrix A defined by (17)

Output: step-size h for the step $t_n \mapsto t_{n+1} = t_n + h$

1. Compute h_{space} :
 - (a) Compute $\mathcal{A}p_n$ and, via a fast wavelet transform, its representation (29).
 - (b) Set $\theta_l = 0$ for all $l = j_1, \dots, j_\eta$.
 - (c) Put $m = s \cdot (\mu_{max} - \eta)$ and set the m largest (in modulus) coefficients to zero. With a fast inverse wavelet transform, compute $\zeta = \sum_{l \notin \mathcal{D}} \theta_l v_l$ where \mathcal{D} is the index set of the discarded terms.
 - (d) Set $h_{\text{space}} = \text{tol} / \|\zeta\|_1$.
2. Compute h_{time} :

(a) Compute

$$(P_W \mathcal{A})^5 p_n = \sum_{i=1}^{\eta} \zeta_i v_{j_i}, \quad (\zeta_1, \dots, \zeta_\eta)^T = A^5 (\beta_1, \dots, \beta_\eta)^T.$$

(b) Set $h_{\text{time}} = C_{\text{safe}} \cdot \left(\frac{720 \cdot \text{tol}}{\|(P_W \mathcal{A})^5 p_n\|_1} \right)^{1/5}$

3. Choose $h = \min\{h_{\text{space}}, h_{\text{time}}\}$.

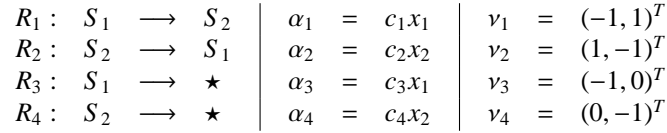
Remark. In a previous version of our code, we did not fix the step-size at the beginning of the time step, but instead changed the step-size during the iterations (steps 2 through 5) based on the new informations gained from the enlargement of the approximation space. However, this strategy was not successful, because the decision which basis elements are chosen depends implicitly on the step-size, such that the basis elements which have been selected in previous iterations are no longer suitable if the step-size has changed.

4. Numerical examples

The following four numerical examples demonstrate the performance of the adaptive wavelet method. The first two examples confirm that the accuracy of our method indeed agrees with the tolerance selected by the user. The third and fourth examples showcase the capability of our approach to solve large, non-trivial problems with bimodal solution profiles.

4.1. Merging Modes

Let us consider two species S_1 and S_2 that interact via the following reaction channels



with rate constants $c_1 = 1.5, c_2 = 0.7, c_3 = 0.7$ and $c_4 = 0.2$. The purpose of this very simple example is to check the behavior of the error with respect to the tolerance selected by the user. This is made possible because the *exact* solution of the corresponding CME is known: all reactions are of *monomolecular* type, and for such systems an explicit formula has been derived in [22].

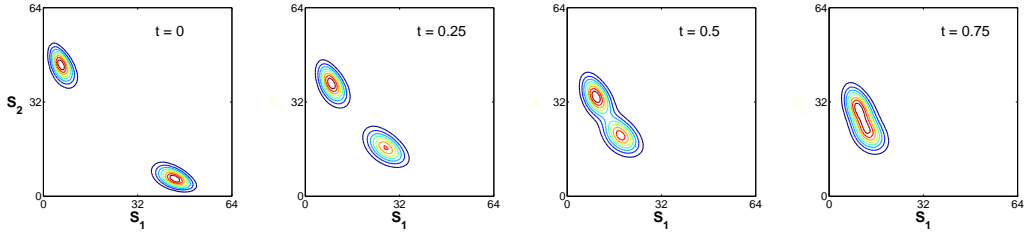


Figure 2: Exact solution of the Merging Modes system at $t = 0, t = 0.25, t = 0.5$ and $t = 0.75$ (from left to right).

For any $x \in \mathbb{N}^2$, $N \in \mathbb{N}$ and any $r = (r_1, r_2)$ with $r_1, r_2 \in [0, 1]$ and $r_1 + r_2 \leq 1$, the multinomial distribution $\mathcal{M}(x, N, r)$ is defined by

$$\mathcal{M}(x, N, r) = \begin{cases} N! \frac{r_1^{x_1}}{x_1!} \frac{r_2^{x_2}}{x_2!} \frac{(1 - r_1 - r_2)^{N - x_1 - x_2}}{(N - x_1 - x_2)!} & \text{if } x_1 + x_2 \leq N \\ 0 & \text{otherwise.} \end{cases}$$

\mathcal{M} is a two-dimensional extension of the well known binomial distribution. For the initial distribution we choose

$$\rho(x) = 0.5 \cdot \mathcal{M}(x, N, r^{(1)}) + 0.5 \cdot \mathcal{M}(x, N, r^{(2)}) \quad (30)$$

with $r^{(1)} = (0.7, 0.1)^T$, $r^{(2)} = (0.1, 0.7)^T$, and $N = 63$. Then, the exact solution of the corresponding CME is

$$p(t, x) = 0.5 \cdot \mathcal{M}(x, N, s^{(1)}(t)) + 0.5 \cdot \mathcal{M}(x, N, s^{(2)}(t)), \quad (31)$$

with

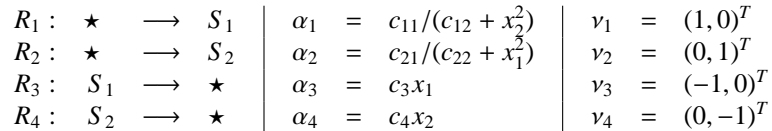
$$s^{(i)}(t) = \exp(tC)r^{(i)}, \quad C = \begin{pmatrix} -(c_1 + c_3) & c_2 \\ c_1 & -(c_2 + c_4) \end{pmatrix}$$

(cf. [22]). Figure 2 shows that $p(t, x)$ consists of two modes which merge to one single peak as time evolves.²

The adaptive wavelet method was applied to this problem on the time interval $[0, 1]$ using *db2* wavelets and four different tolerances. In the left panel of Figure 3 we plot the error of the adaptive wavelet method in the 1-norm by comparing each of the approximations with the explicitly derived solution. The plot illustrates that for tolerances up to $\text{tol} = 10^{-3}$ the error estimator described in (28) works well and the error is almost always below the chosen tolerance. For smaller tolerances, however, the selection of the step-size is too optimistic, i.e. the steps are not small enough. We remark that this behavior appears only for tolerances that are going to be used for small problems. An easy fix is to use a safety factor C_{safe} in the second term in (28), and the result is illustrated in the right panel of Figure 3. It is also important to mention that the 1-norm scales with the state space, so for bigger problems, a tolerance of 10^{-1} or 10^{-2} provides a sufficiently good accuracy. As additional information, the evolution of the step size and the number of basis elements used by the runs without the safety factor are plotted in Figure 4.

4.2. Genetic Toggle Switch

In this example, we investigate a pair of mutually repressing genes, where the two competing species S_1 and S_2 each inhibits the transcription of its opponent. The reaction channels are



²Because of the discrete nature of the solution of the CME, contour or mesh plots are somewhat misleading, since the probability distribution is only defined at discrete points $x \in \mathbb{N}^d$. Such plots, however, provide a much clearer picture of the solution profile than other visualization methods.

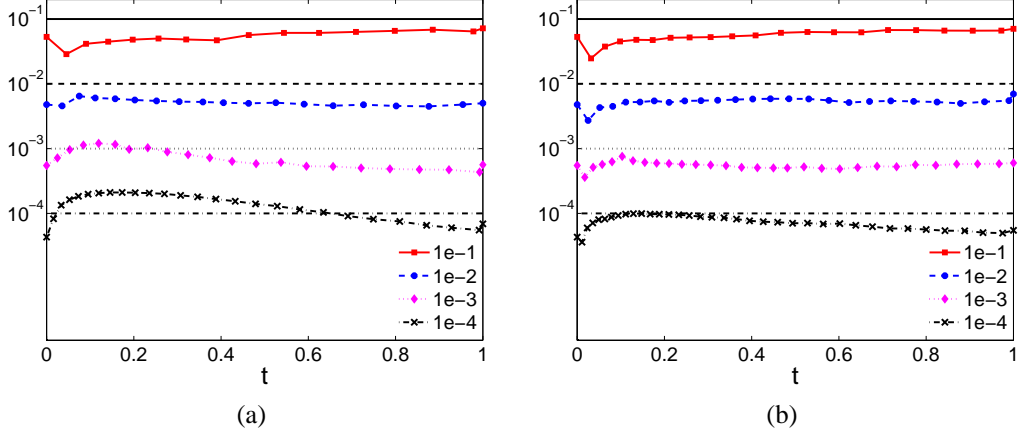


Figure 3: Left panel (a): Error of the adaptive wavelet approximation of the Merging Modes problem for $tol_1 = 10^{-1}$ (square), $tol_2 = 10^{-2}$ (circle), $tol_3 = 10^{-3}$ (diamond) and $tol_4 = 10^{-4}$ (cross). The error was computed in the 1-norm by comparing each of the approximations with the exact solution. Left panel (b): Error of the adaptive wavelet approximation for $tol_1 = 10^{-1}, 10^{-2}, 10^{-3}$ and 10^{-4} using a safety factor $C_{safe} = 0.7$ for h_{time} .

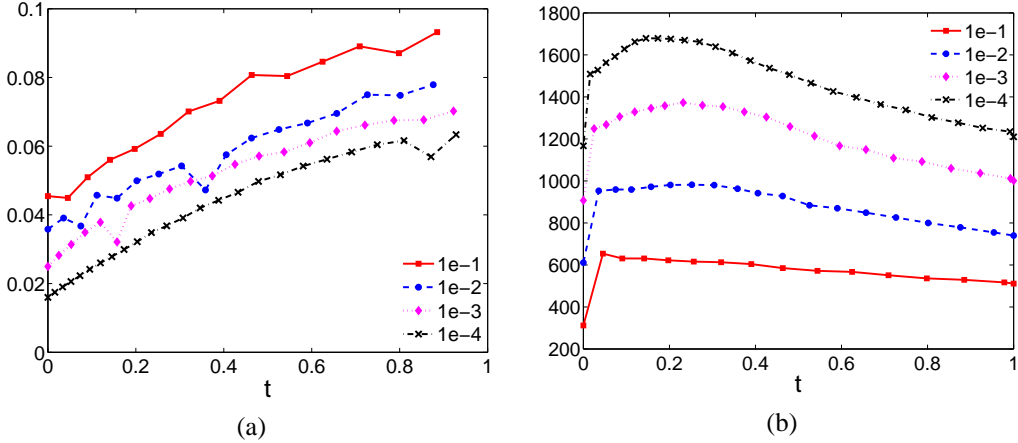


Figure 4: Left panel (a): Evolution of the step-size h for the Merging Modes problem without the safety factor, using $tol_1 = 10^{-1}$ (solid), $tol_2 = 10^{-2}$ (dashed), $tol_3 = 10^{-3}$ (dotted) and $tol_4 = 10^{-4}$ (dash-dot). Right panel (b): Number of basis elements used in each step to compute the approximation for $tol_1 = 10^{-1}$ (solid), $tol_2 = 10^{-2}$ (dashed), $tol_3 = 10^{-3}$ (dotted) and $tol_4 = 10^{-4}$ (dash-dot).

with parameters $c_{11} = c_{21} = 10$, $c_{12} = c_{22} = 30$ and $c_3 = c_4 = 0.017$. If copies of S_2 are present in abundance, then the propensity function for reaction R_1 almost vanishes, which inhibits the transcription of new copies of S_1 . However, over sufficiently long time intervals, stochastic fluctuations can cause an increase in the copy-numbers of S_1 , meaning that the production of S_2 will be inhibited instead, and leading to a *switch* in the roles of S_1 and S_2 . Consequently, the

solution of the CME develops two peaks that correspond to the two possible scenarios. Reactions R_3 and R_4 model the decay of the two competing species.

The corresponding CME was solved by the adaptive wavelet method on the time interval $[0, 500]$ using *db3* wavelets. As initial distribution a “discrete Gaussian”,

$$p(0, x) = \gamma \cdot \exp(-(x - \mu)^T C (x - \mu)), \text{ for all } x \in \Omega_\xi,$$

$$C = \begin{pmatrix} 10000 & 0 \\ 0 & 10000 \end{pmatrix}$$

centered at $\mu = (20, 18)$ was chosen; the normalization constant γ was determined via the condition $\sum_{x \in \Omega_\xi} p(0, x) = 1$. In this example, the truncated state space $\Omega_{32,32}$ was small enough such that a reference solution could be obtained with the MATLAB routine *ode15s*. Three different runs of the adaptive wavelet method using the same parameters but different tolerances were performed. In the left panel of Figure 5 the error of the adaptive wavelet approximation for each of the three tolerances is shown. The error was computed in the 1-norm by comparing each of the approximations with a reference solution obtained using the routine *ode15s*. The plot reveals that the error usually lies below the chosen tolerance and thus the method almost always provides an approximation of the exact solution at the desired accuracy. In the right panel, the time evolution of the step-size h corresponding to the different tolerances is shown. As expected, the adaptive method selects larger time steps for low tolerances, while higher tolerances imply the use of smaller step-sizes. Moreover, small time steps are only required in the stiff transient phase at the beginning of the time interval, which shows that our adaptive step-size control is clearly more efficient than time integration with a fixed step-size.

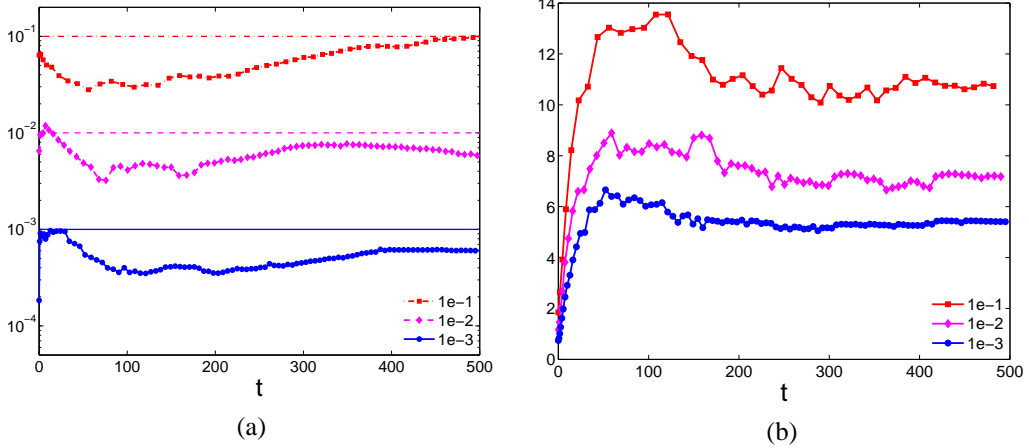
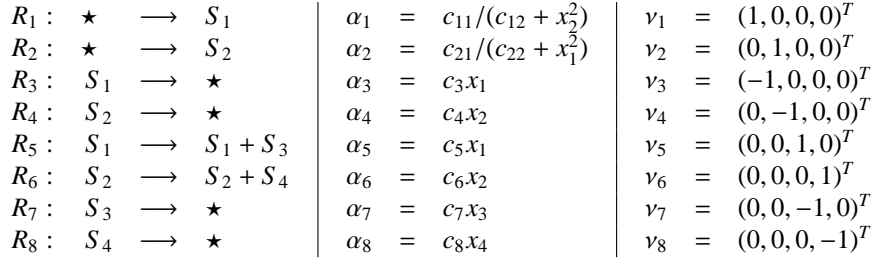


Figure 5: Left panel (a): Error of the adaptive wavelet approximation of the toggle switch for $tol_1 = 0.1$ (square), $tol_2 = 0.01$ (diamond) and $tol_3 = 0.001$ (circle). The error was computed in the 1-norm by comparing each of the approximations with the reference solution. Right panel (b): Evolution of the step-size h for the toggle switch solved by the adaptive wavelet method with $tol_1 = 0.1$ (square), $tol_2 = 0.01$ (diamond) and $tol_3 = 0.001$ (circle).

4.3. Extended Toggle Switch

As a third more challenging example, we consider another genetic toggle switch, which consists of two mutually repressing gene products, S_1 and S_2 , that express two proteins, denoted by S_3 and S_4 respectively. The interactions between these four species ($d = 4$) are modeled by the following reaction system:



In addition to reactions R_1 through R_4 which have been discussed in the previous subsection, four more reactions have been added to the model. R_7 and R_8 model the decay of the respective species, while the expression of proteins is described by reactions R_5 and R_6 . The parameters for the reaction channels are

$$c_{11} = c_{21} = 10, \quad c_{12} = c_{22} = 30, \quad c_3 = c_4 = 0.017, \quad c_5 = c_6 = c_7 = c_8 = 0.01,$$

with the initial distribution being a “discrete Gaussian” with a small variance, centered at $\mu = (20, 18, 22, 5)$, which closely resembles a delta peak located at μ . The number of degrees of freedom in this example is 2^{20} , the state space being $32 \times 32 \times 32 \times 32$ ($d = 4$). The corresponding CME was solved by the adaptive wavelet method on the time interval $[0, 500]$, with the method being configured to use $\tau_{01} = 0.5$ in the 1-norm. This value seems to be unreasonably large, but since the 1-norm scales with the size of the state space, this choice will in fact provide a very good accuracy. An equally distributed error ε with $\|\varepsilon\|_1 = 0.5$ would, for example, correspond to a maximal error of $\|\varepsilon\|_\infty = 0.5/2^{20} \approx 4.77 \cdot 10^{-7}$.

The method was configured to keep a minimum of 5000 of the largest coefficients at the end of each time step, while the total number of elements that could be used within the algorithm was not allowed to exceed 6000. Hence, the solution was approximated using only 0.47% of the total number of 1,048,576 degrees of freedom. New basis elements were proposed in batches of 250 elements each and the *db3* wavelet basis was again chosen to approximate the solution.

The time evolution of the CME solution is shown in Figure 8. As the full distribution is a four-dimensional object, we only plot the most relevant 2D marginal distributions at different times. The first two columns depict mesh and contour plots of the marginal distribution of the gene products $S_1 - S_2$ at different times, while in the third column, a contour plot for the marginal distribution of the proteins $S_3 - S_4$ is shown. Pronounced bi-modality is evident at $t = 500$ in both marginal distributions.

The change in step-size h for the integrator is plotted in panel (a) of Figure 6, and the 1-norm of the residual is shown in panel (b). The stiffness of the problem is clearly visible in Figure 8, and the evolution of the time step conforms to the expectation that small step-sizes are selected in the initial phase, while larger ones are possible as the distribution approaches the steady state.

We remark that by eliminating from the model the reactions involving the proteins S_3 and S_4 , i.e., R_5 through R_8 , we obtain the simplified 2D toggle switch presented in Section 4.2. The

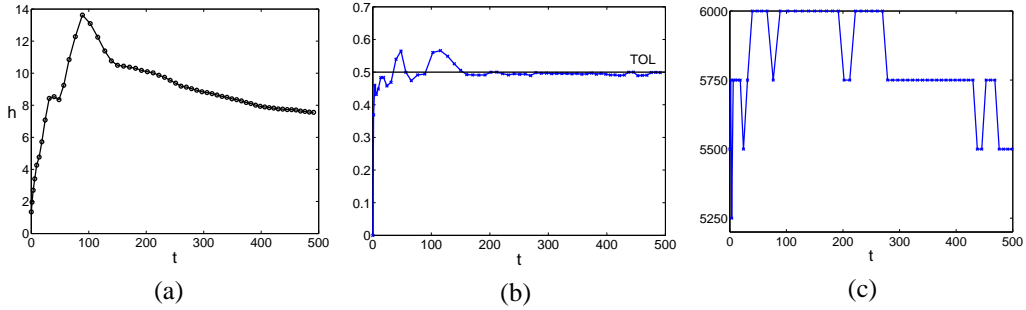
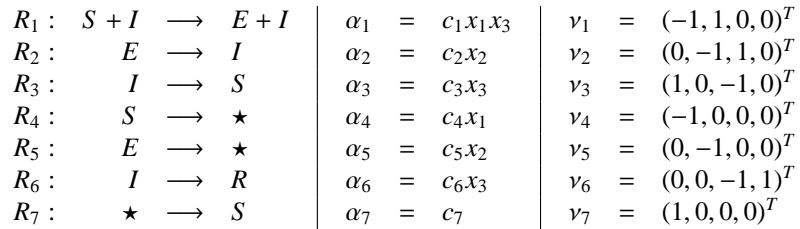


Figure 6: Left panel (a): Evolution of step-size h for the $4D$ toggle switch. Middle panel (b): Evolution of the 1-norm (scaled) of the residual for the same problem. Right panel (c): Number of basis elements used in each step to compute the approximation.

solution of this smaller problem agrees with the marginal distribution in the $S_1 - S_2$ plane of the $4D$ model, and as such can be used as a sort of reference solution, because the truncated state space $\Omega_{32,32}$ of the simplified problem is small enough such that it is possible to compute a reference solution with the MATLAB routine *ode15s*. In Figure 7, we use this property to illustrate the need for higher-order wavelet basis when the number of degrees of freedom is large. Firstly, the Haar basis is used to compute an approximation using the adaptive wavelet method for the $4D$ toggle switch. The marginal distribution in the $S_1 - S_2$ plane is shown at time $t = 500$ in the leftmost panel. The middle panel displays the results obtained using the same parameters for the solver, but this time employing the *db3* wavelet basis. The “reference” solution computed using MATLAB’s *ode15s* on the simplified $2D$ problem is shown in the right panel. It is immediately clear that for problems of a certain size, the Haar wavelet basis used in [20] is no longer adequate as the number of basis elements needed would simply be too high. This factor would then drive the computational cost above reasonable levels. In contrast, the possibility to use the entire Daubechies wavelet family (which includes Haar) increases the flexibility of the adaptive wavelet method allowing the efficient numerical treatment of a variety of problems.

4.4. Infectious diseases

The SEIR is an epidemic model used to describe the spread of communicable diseases within a population (see [19] for details). The population is split into four classes ($d = 4$), namely individuals susceptible to become infected with the disease (S), exposed individuals (E) that are infected but not yet contagious, infectious individuals (I) and individuals that have recovered (R), and in the process acquired immunity to the disease. The sub-populations of the model interact via the following reaction channels:



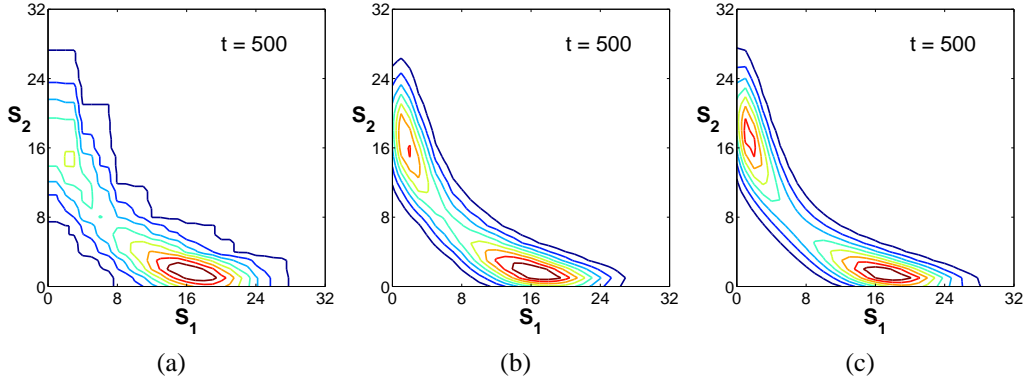


Figure 7: Comparison between approximations for the 4D toggle switch obtained using Haar basis (a), Daubechies *db3* wavelet basis (b) and Matlab's *ode15s* on the simplified 2D problem (c)

Reaction R_1 models the process through which susceptible individuals become infected by having contact with infectious ones. The infected individuals first enter the latent phase of the disease becoming members of the E class, and can become infectious themselves after the incubation period via reaction R_2 . The temporary recovery of infected individuals can occur via reaction R_3 . Reactions R_4 and R_5 describe the death of susceptible and exposed individuals, whereas reaction R_7 represents new arrivals that are prone to becoming infected. Reaction R_6 describes the recovering process of infectious individuals, that also acquire immunity to the disease. We assume that the inflow of susceptible individuals via reaction R_7 is constant and is independent of the current size of the population. We are interested in the scenario where the disease starts with only a few infected individuals. This variant of the model is stochastic as the question whether the disease quickly spreads to a large section of the population or disappears at some early stage is dependent on the fate of these first infectious individuals. In our example the parameters for the reaction channels are

$$c_1 = 0.1, \quad c_2 = 0.5, \quad c_3 = 1, \quad c_4 = c_5 = c_6 = 0.01, \quad c_7 = 0.4,$$

and as initial distribution, a "discrete Gaussian" centered at $\mu = (50, 4, 0, 0)$ was considered.

Figure 9 shows the time evolution of the probability distribution for our SEIR model. For the marginal distribution in the S-E plane both contour and mesh plots are shown (left and middle column, from top to bottom). The rightmost column shows a contour plot for the marginal distribution in the S-I plane. During the course of the simulation, the marginal distribution in the S-E plane splits up into two distinct peaks. The peak located at $(50, 0)$ depicts the scenario in which the first few infectious individuals have either died or recovered before their number reached a critical mass and therefore the disease extinguished itself after some time. If the opposite happens, and the infection spreads fast enough during the initial phase, then the system will eventually reach a state where most of the population is infected, as indicated by the second peak located at $(11, 27)$. Similar bimodalities appear in the marginal distributions of the S-I plane (left column) and E-I plane (data not shown).

The fact that the solution is multi-modal and the peaks are located far apart poses no significant challenges to the adaptive wavelet method. In the last row of panels from Figure 9, the

non-smooth character of the solution can be clearly observed. At time $t = 7$ the solution vanishes close to the S -axis but does not vanish *on* the axis itself. This local non-smoothness will pose problems to methods which assume a certain regularity of the solution. Although wavelets are best suited for the approximation of sufficiently smooth signals, the method is also able to handle such difficult scenarios.

Between $t = 3$ and $t = 5$ the solution profile for SEIR conforms to a rather thin line that is not parallel to any of the axes. Methods which represent the solution in terms of global tensor products (e.g., the method from [23]) would need too many degrees of freedom to achieve an usable accuracy. The adaptive wavelet method suffers from no such drawbacks, because the elements of the wavelet basis used are *local* tensor products with small support.

As before, the adaptive wavelet method was used with the *db3* wavelet basis to approximate the solution of the corresponding CME. The choice of wavelet basis was motivated by the desire to have good compression properties while keeping the support reasonably small. In order to cover the time interval $[0, 7]$, 122 steps of the algorithm were required. The method was configured to use $\text{tol} = 0.61$ in the 1-norm, and the iteration for solving the linear system was stopped if this tolerance was met or if the total number of basis elements exceeded 6500. At the end of each step, only the largest 6000 coefficients were kept which corresponds to 0.57% out of the total number of 2^{20} degrees of freedom. New basis elements were added in batches of 250 elements each and solving the linear system (19) was accomplished via GMRES with restarts and a tolerance of $5 \cdot 1e^{-4}$.

From a numerical point of view, the biggest limiting factor in computing the solution of the CME for this and the previous problem is the huge state space with more than 1,000,000 states. As can be clearly seen in the panels of Figures 8 and 9, most of these states are never populated throughout the time evolution, which means that the subset of *essential* states is actually smaller. However, this information is of little practical use, because we only know which states can be ignored *a posteriori*. As the adaptive wavelet method is specifically designed to find the *essential* degrees of freedom, it is particularly suited to deal with these type of problems.

Acknowledgment

It is our pleasure to thank Andreas Rieder for his valuable advice on wavelet analysis, and the anonymous referees for their comments. The Research Group “Numerical methods for high-dimensional systems” received financial support by the “Concept for the Future” of Karlsruhe Institute of Technology within the framework of the German Excellence Initiative. Moreover, support by the DFG priority programme SPP 1324 “Mathematische Methoden zur Extraktion quantifizierbarer Information aus komplexen Systemen” is acknowledged.

References

- [1] K. Burrage, M. Hegland, S. MacNamara, and R. B. Sidje. A Krylov-based finite state projection algorithm for solving the chemical master equation arising in the discrete modelling of biological systems. In A.N.Langville and W.J.Stewart, editors, *Markov Anniversary Meeting: An international conference to celebrate the 150th anniversary of the birth of A.A. Markov*, pages 21 – 38. Bosen Books, 2006.
- [2] A. Cohen. *Numerical Analysis of wavelet methods*, volume 32 of *Studies in Mathematics and its applications*. Elsevier, 2003.
- [3] A. Cohen, W. Dahmen, and R. DeVore. Adaptive wavelet methods for elliptic operator equations: Convergence rates. *Math. Comput.*, 70(233):27–75, 2001.
- [4] A. Cohen, W. Dahmen, and R. DeVore. Adaptive wavelet methods. II: Beyond the elliptic case. *Found. Comput. Math.*, 2(3):203–245, 2002.

- [5] W. Dahmen. Wavelets and multiscale methods for operator equations. *Acta Numerica*, 6:55–228, 1997.
- [6] W. Dahmen. Wavelet methods for PDEs – some recent developments. *J. Comput. Appl. Math.*, 128, 133-185, 2001.
- [7] I. Daubechies. *Ten lectures on wavelets*. CBMS-NSF Regional Conference Series in Applied Mathematics. 61. Philadelphia, PA: SIAM, Society for Industrial and Applied Mathematics., 1992.
- [8] P. Deuffhard, W. Huisinga, T. Jahnke, and M. Wulkow. Adaptive discrete Galerkin methods applied to the chemical master equation. *SIAM J. Sci. Comput.*, 30(6):2990–3011, 2008.
- [9] S. Engblom. Galerkin spectral method applied to the chemical master equation. *Commun. Comput. Phys.*, 5(5):871–896, 2009.
- [10] S. Engblom. Spectral approximation of solutions to the chemical master equation. *J. Comput. Appl. Math.*, 229(1):208–221, 2009.
- [11] L. Ferm and P. Lötstedt. Adaptive solution of the master equation in low dimensions. *Appl. Numer. Math.*, 59(1):187–204, 2009.
- [12] L. Ferm, P. Lötstedt, and A. Hellander. A hierarchy of approximations of the master equation scaled by a size parameter. *SIAM J. Sci. Comput.*, 34:127–151, 2008.
- [13] D. T. Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *J. Comput. Phys.*, 22:403–434, 1976.
- [14] D. T. Gillespie. A rigorous derivation of the chemical master equation. *Physica A*, 188:404–425, 1992.
- [15] E. Hairer and G. Wanner. *Solving ordinary differential equations. II: Stiff and differential-algebraic problems*. Number 14 in Springer Series in Computational Mathematics. Springer, Berlin, 2nd rev. ed. edition, 1996.
- [16] M. Hegland, C. Burden, L. Santoso, S. MacNamara, and H. Booth. A solver for the stochastic master equation applied to gene regulatory networks. *J. Comput. Appl. Math.*, 205:708–724, 2007.
- [17] M. Hegland, A. Hellander, and P. Lötstedt. Sparse grids and hybrid methods for the chemical master equation. *BIT*, 48:265–284, 2008.
- [18] A. Hellander and P. Lötstedt. Hybrid method for the chemical master equation. *J. Comput. Phys.*, 227:100–122, 2007.
- [19] H. W. Hethcote. The mathematics of infectious diseases. *SIAM Review*, 42(4):599–653, 2000.
- [20] T. Jahnke. An adaptive wavelet method for the chemical master equation. *SIAM Journal on Scientific Computing*, 31(6):4373–4394, 2010.
- [21] T. Jahnke and S. Galan. Solving chemical master equations by an adaptive wavelet method. In T. E. Simos, G. Psihoyios, and C. Tsitouras, editors, *Numerical Analysis and applied mathematics: International Conference on Numerical Analysis and Applied Mathematics 2008. Psalidi, Kos, Greece, 16-20 September 2008*, volume 1048 of *AIP Conference Proceedings*, pages 290–293, 2008.
- [22] T. Jahnke and W. Huisinga. Solving the chemical master equation for monomolecular reaction systems analytically. *J. Math. Biol.*, 54(1):1-26, 2007.
- [23] T. Jahnke and W. Huisinga. A dynamical low-rank approach to the chemical master equation. *Bulletin of Mathematical Biology*, 70(8):2283–2302, 2008.
- [24] S. MacNamara, A. M. Bersani, K. Burrage and R. B. Sidje. Stochastic chemical kinetics and the total quasi-steady-state assumption: Application to the stochastic simulation algorithm and chemical master equation. *J. Chem. Phys.*, 129(9):95–105, 2008.
- [25] S. MacNamara, K. Burrage, and R. B. Sidje. Multiscale modeling of chemical kinetics via the master equation. *Multiscale Model. Simul.*, 6(4):1146–1168, 2008.
- [26] S. Mallat. *A wavelet tour of signal processing. 3rd ed.* Amsterdam: Elsevier., 2009.
- [27] B. Munsky and M. Khammash. The finite state projection algorithm for the solution of the chemical master equation. *J. Chem. Phys.*, 2006.
- [28] T. von Petersdorff and Ch. Schwab. Numerical solution of parabolic equations in high dimensions. *M2AN, Math. Model. Numer. Anal.*, 38, No. 1, 93-127, 2004.
- [29] Ch. Schwab and R. Stevenson. Space-time adaptive wavelet methods for parabolic evolution problems. *Math. Comp.*, 78, 1293-1318, 2009.
- [30] P. Sjöberg, P. Lötstedt, and J. Elf. Fokker-Planck approximation of the master equation in molecular biology. *Comput. Visual. Sci.*, 12(1):37–50, 2009.

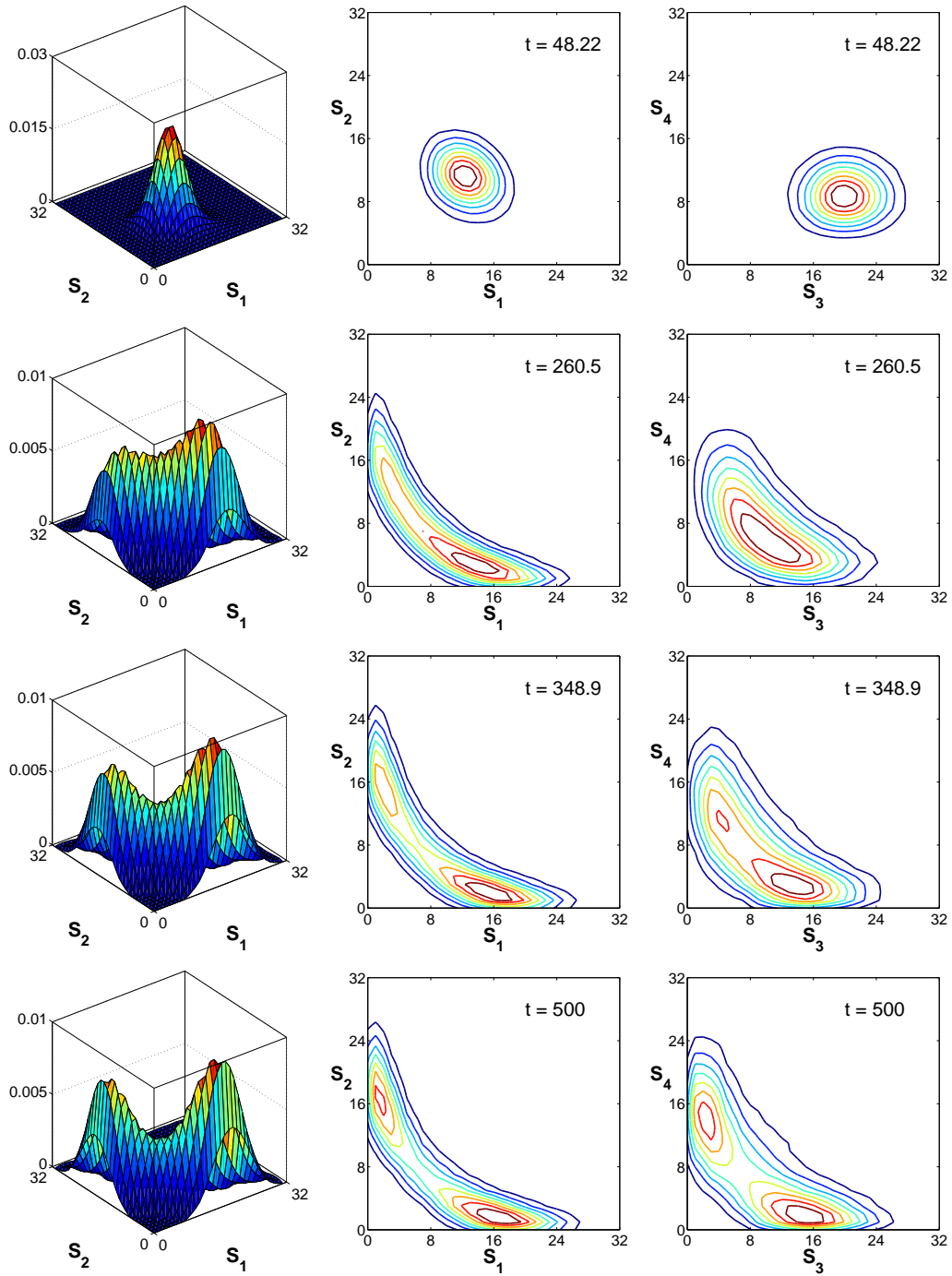


Figure 8: Marginal distribution of the 4D toggle switch model at different times. Surf plot (first column) and contour plots (second and third columns) of the approximation obtained with the adaptive wavelet method.

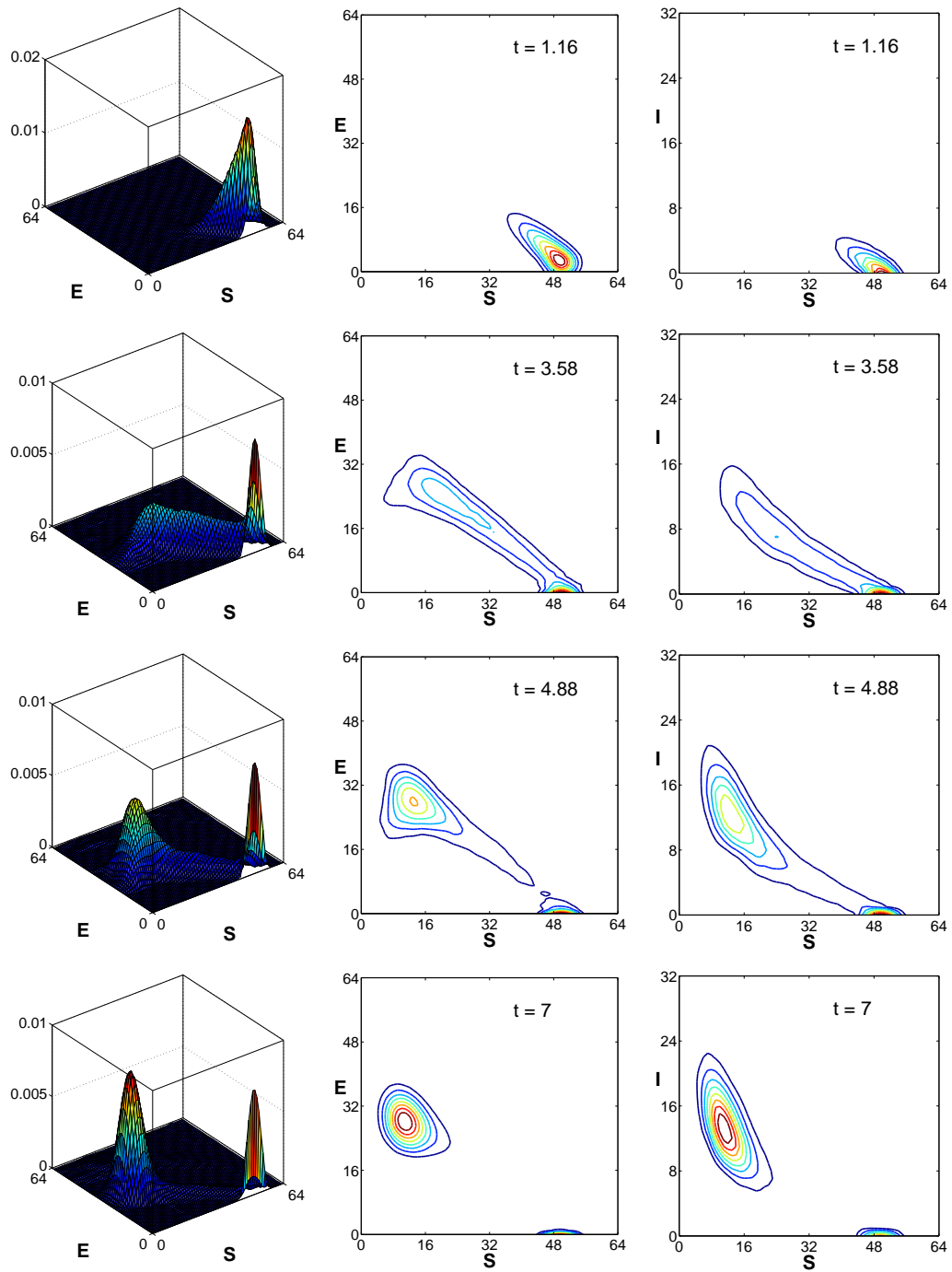


Figure 9: Marginal distributions of the stochastic SEIR model at different times. Surf plot (first column) and contour plots (second and third columns) of the approximation obtained with the adaptive wavelet method.